EmbodMocap: In-the-Wild 4D Human-Scene Reconstruction for Embodied Agents

Supplementary Material

1. More Details of EmbodMocap

1.1. Capture technique

The primary capture technique involves two photographers, each holding an iPhone in a vertical orientation. The photographers are required to maintain a certain angle relative to each other while following the performer. To achieve optimal triangulation during post-processing, the angle between the two cameras should ideally fall within the range of 60 to 120 degrees.

This configuration not only enhances the accuracy of triangulation but also ensures the capture of the performer from multiple perspectives, providing diverse viewpoint information for keypoint detection. Additionally, the photographers should aim to keep the cameras in motion to dynamically adjust their positions and minimize occlusion caused by objects in the environment.

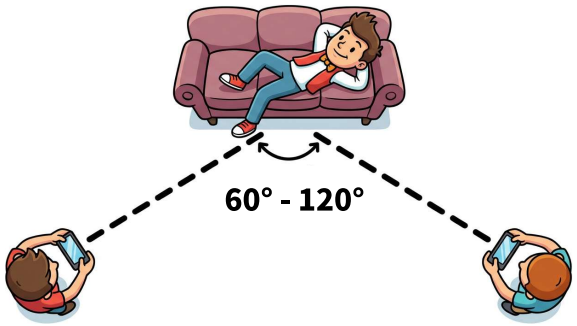


Figure 1. Capture technique.

1.2. Human Labor Analysis

Temporal Synchronization. This step only needs the operator to identify and input the frame indices where the laser pointer's spot disappears into a `.xlsx` file. Typically, this process takes only about 1 minute per sequence.

Skill Segmentation. Skill segmentation is only required when training physical interaction skills. The operator annotates each skill's category, start, and end times based on the video, typically taking 0.5 to 2 minutes per sequence.

Contact Label & Optimization. In the main text, we mention that the alignment between our sequence and the scene coordinate system relies on photometric (COLMAP, pixel tracking) and geometric constraints (chamfer distance). However, this can sometimes result in alignment errors of a few centimeters, primarily due to depth inaccuracies in COLMAP's sparse keypoints and depth errors from the iPhone sensor. To address this issue, we propose an optional post-processing solution. During data capture, we place markers in the scene and instruct the performer to begin walking from a designated marker and stop on another at the end of the sequence, standing still on the same marker. Annotating contact frame indices costs 1-2 minutes for each sequence. These markers serve as fixed reference points for alignment. In post-processing, we observe the corresponding marker positions on the reconstructed mesh and record their 3D coordinates, along with the frame indices where the performer stands on the markers. Using this information, we optimize a rigid transformation to align the center of the performer's feet at the specified frame indices to the 3D coordinates of the markers.

Since SpectacularAI could generate Z-up metric-scaled camera matrices, we define the rigid transformation in the xy-plane, defined by a rotation angle ϕ_c about the z-axis and a translation $\boldsymbol{T}_c$. This can be represented by a homogeneous transformation matrix $\boldsymbol{M}$:

$$M = \begin{bmatrix} \boldsymbol{R}(\phi_c) & \boldsymbol{T}_c \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \cos(\phi_c) & -\sin(\phi_c) & 0 & t_x \\ \sin(\phi_c) & \cos(\phi_c) & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

This matrix transform the center of lowest point on both feet to match the annotate marker. To robustly solve for the transformation parameters, we employ a gradient descent optimization, constrained by a minimizing a contact loss to match the contact marker:

$$\mathcal{L}_{\text{contact}} = \frac{1}{N_c} \sum_{i \in \mathcal{C}} \left(\min_z(\mathcal{V}^{(i)}) - c_z^{(i)} \right)^2 \tag{2}$$

For SMPL parameters, the global orientation is updated as $\boldsymbol{\theta}'^g = \boldsymbol{R}_c \boldsymbol{\theta}^g$. For translation, the pelvis's world position is transformed as $\boldsymbol{P}'_w = \boldsymbol{R}_c \boldsymbol{P}_w + \boldsymbol{T}_c$. Re-evaluating the SMPL model with $\boldsymbol{\theta}'^g$ gives the local pelvis offset $\boldsymbol{P}'_l$, and the updated translation is $\boldsymbol{\gamma}' = \boldsymbol{P}'_w - \boldsymbol{P}'_l$.

The updated camera rotation and translation are computed as $\boldsymbol{R}'_v = \boldsymbol{R}_v \boldsymbol{R}c^T$ and $\boldsymbol{T}'_v = \boldsymbol{T}_v - \boldsymbol{R}_v \boldsymbol{R}c^T \boldsymbol{T}_c$, ensuring alignment and consistency of the scene representation.

2. More Details of Monocular Human-Scene Reconstruction Pipeline

Our monocular reconstruction baseline is a modular pipeline for reconstructing 3D human pose and scene geometry from monocular video, combining two independent

Figure 2. An example in finding the contact marker in software (e.g., Meshlab) and corresponding keyframe index(the frames selected here are just for demo).

modules: π^3 for camera trajectory prediction and scene point cloud reconstruction, and VIMO for SMPL-based human pose estimation. To process long video sequences, π^3 divides frames into overlapping chunks, where each chunk independently predicts camera poses $\boldsymbol{T}_v \in \mathbb{R}^{T\times4\times4}$ and local point clouds $\boldsymbol{P}_{\text{local}} \in \mathbb{R}^{T\times H\times W\times3}$. To align these chunks into a global coordinate system, Procrustes analysis is applied to the overlapping regions of adjacent chunks. Given two point clouds $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{N\times3}$, the alignment minimizes the error:

$$\min_{s,\boldsymbol{R},\boldsymbol{t}} \|\boldsymbol{Y} - (s\boldsymbol{R}\boldsymbol{X} + \boldsymbol{t})\|_F^2, \tag{3}$$

where s is the scale, $\boldsymbol{R}$ is the rotation matrix, and $\boldsymbol{t}$ is the translation vector. Using SVD, the optimal alignment parameters are computed as:

$$\boldsymbol{R} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^\top, \quad s = \frac{\text{trace}(\boldsymbol{Y}_c^\top \boldsymbol{R}\boldsymbol{X}_c)}{\text{trace}(\boldsymbol{X}_c^\top \boldsymbol{X}_c)}, \quad \boldsymbol{t} = \bar{\boldsymbol{Y}} - s\boldsymbol{R}\bar{\boldsymbol{X}}, \tag{4}$$

where $\boldsymbol{X}_c, \boldsymbol{Y}_c$ are the centered point clouds, and $\boldsymbol{V}, \boldsymbol{U}$ are derived from the SVD of the covariance matrix $\boldsymbol{H} = \boldsymbol{X}_c^\top \boldsymbol{Y}_c$. After chunk alignment, VIMO predicts SMPL parameters $(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\beta})$, where $\boldsymbol{\theta} \in \mathbb{R}^{T\times72}$ represents joint rotations, $\boldsymbol{\gamma} \in \mathbb{R}^{T\times3}$ is the root translation, and $\boldsymbol{\beta} \in \mathbb{R}^{10}$ defines body shape. Using a weak perspective camera model, SMPL vertices are projected onto the image plane as:

$$\boldsymbol{x}_{\text{img}} = s\boldsymbol{x}_v + \boldsymbol{t} \tag{5}$$

where s is the scaling factor proportional to $1/z$. To resolve scale ambiguity, the pipeline estimates a metric scale by matching the predicted depths of SMPL vertices z_{SMPL} (in meters) with the depths of Pi3's point cloud z_{Pi3} (in arbitrary units) on some sampled points. The scale factor is computed as:

$$s = \text{median}\left(\frac{z_{\pi^3}}{z_{\text{SMPL}}}\right), \tag{6}$$

The point clouds and SMPL global orientation and translation are transformed to the world coordinate system with $\boldsymbol{R}, \boldsymbol{t}$ following the same formula as Sec. 1.2.

3. More Details of Human-Object Interaction Skills

3.1. Follow Skill

Definition. The path following task requires the simulated character to move along a predefined 2D trajectory. A trajectory is represented as $\tau = \{x_{0.1}^\tau, x_{0.2}^\tau, \ldots, x_{T-0.1}^\tau, x_T^\tau\}$, where $x_{0.1}^\tau$ denotes a 2D waypoint at simulation time $0.1s$, and T is the episode length. For this task, T is set to $10s$. The character is expected to follow the trajectory τ as accurately as possible.

Task Observation. At each simulation time step t, the character observes 10 future waypoints sampled over the next $1.0s$: $\{x_t^\tau, x_{t+0.1}^\tau, \ldots, x_{t+0.8}^\tau, x_{t+0.9}^\tau\}$. These waypoints are sampled at intervals of $0.1s$ using linear interpolation from the trajectory τ. The 2D coordinates of these waypoints form the task observation $g_t^f \in \mathbb{R}^{2\times10}$.

Task Reward. The reward for this task, r_t^f, is computed based on the distance between the character's current 2D root position, $x_t^{\text{root\_2d}}$, and the target waypoint, x_t^τ. The reward is defined as:

$$r_t^f = \exp\big(-2.0\|x_t^{\text{root\_2d}} - x_t^\tau\|^2\big). \tag{7}$$

3.2. Sit Skill

Definition. The sitting task requires the character to position its root joint at a target 3D sitting location on an object surface. The target position is defined as 10 cm above the center of the top surface of the chair seat.

Task Observation. The observation $g_t^s \in \mathbb{R}^{38}$ includes the 3D target sitting position $\in \mathbb{R}^3$, the 3D root position $\in \mathbb{R}^3$, the root rotation $\in \mathbb{R}^6$, the 2D front-facing direction $\in \mathbb{R}^2$, and the positions of eight corner points of the object's bounding box $\in \mathbb{R}^{3\times8}$.

Task Reward. The sitting task reward r_t^s encourages the character to minimize the distance between its 3D root position, x_t^{root}, and the target sitting position, x_t^{tar}. It is defined as:

$$r_t^s = \begin{cases} 0.7\, r_t^{\text{near}} + 0.3\, r_t^{\text{far}}, & \|x_t^{\text{obj\_2d}} - x_t^{\text{root\_2d}}\| > 0.5, \\ 0.7\, r_t^{\text{near}} + 0.3, & \text{otherwise}, \end{cases} \tag{8}$$

where r_t^{far} and r_t^{near} are defined as:

$$r_t^{\text{far}} = \exp\big(-2.0\|1.5 - d_t^* \cdot \dot{x}_t^{\text{root\_2d}}\|^2\big), \tag{9}$$

$$r_t^{\text{near}} = \exp\big(-10.0\|x_t^{\text{tar}} - x_t^{\text{root}}\|^2\big). \tag{10}$$

Here, $x_t^{\text{obj\_2d}}$ is the 2D position of the object's root, $\dot{x}_t^{\text{root\_2d}}$ is the 2D linear velocity of the character's root, and d_t^* is a horizontal unit vector pointing from $x_t^{\text{root\_2d}}$ to $x_t^{\text{obj\_2d}}$.

3.3. Climb Skill

Definition. The climbing task requires the character to place its root joint at a target 3D climbing position on a given object. The target position is set 94 cm above the center of the top surface of the object.

Task Observation. The observation $g_t^m \in \mathbb{R}^{27}$ includes the 3D target root position $\in \mathbb{R}^3$ and the 3D coordinates of eight corner points of the object's bounding box $\in \mathbb{R}^{3\times 8}$.

Task Reward. The climbing task reward r_t^m minimizes the 3D distance between the character's root, x_t^{root}, and the target location, x_t^{tar}. The reward is defined as:

$$r_t^m = \begin{cases} 0.5\, r_t^{\text{near}} + 0.2\, r_t^{\text{far}}, & \|x_t^{\text{obj\_2d}} - x_t^{\text{root\_2d}}\| > 0.7, \\ 0.5\, r_t^{\text{near}} + 0.2 + 0.3\, r_t^{\text{foot}}, & \text{otherwise}, \end{cases} \tag{11}$$

where r_t^{near}, r_t^{far}, and r_t^{foot} are defined as:

$$r_t^{\text{near}} = \exp\big(-10.0\|x_t^{\text{tar}} - x_t^{\text{root}}\|^2\big), \tag{12}$$

$$r_t^{\text{far}} = \exp\big(-2.0\|1.5 - d_t^* \cdot \dot{x}_t^{\text{root\_2d}}\|^2\big), \tag{13}$$

$$r_t^{\text{foot}} = \exp\big(-50.0\|(x_t^{\text{tar\_h}} - 0.94) - x_t^{\text{foot\_h}}\|^2\big). \tag{14}$$

Here, $x_t^{\text{tar\_h}}$ is the height of the target root position, $(x_t^{\text{tar\_h}} - 0.94)$ represents the height of the top surface of the target object in world coordinates, and $x_t^{\text{foot\_h}}$ is the mean height of the character's feet. The reward r_t^{foot} encourages the character to lift its feet and is crucial for successful climbing.

3.4. Lie Skill

Definition. The lying task requires the character to position its root joint at a target 3D lying position on an object, typically centered on the object's surface. The character must first approach a designated standing point before transitioning into the lying position.

Task Observation. The observation $g_t^l \in \mathbb{R}^{38}$ includes the 3D target lying position $\in \mathbb{R}^3$, the 3D root position $\in \mathbb{R}^3$, the root rotation $\in \mathbb{R}^6$, the 2D front-facing direction $\in \mathbb{R}^2$, and the positions of eight corner points of the object's bounding box $\in \mathbb{R}^{3\times 8}$. It also includes the chosen standing point $\in \mathbb{R}^3$.

Task Reward. The lying reward r_t^l combines rewards for approaching the standing point and accurately lying down:

$$r_t^l = \begin{cases} 0.6\, r_t^{\text{near}} + 0.4\, r_t^{\text{far}}, & \|x_t^{\text{root}} - x_t^{\text{tar}}\| > 1.5, \\ r_t^{\text{near}}, & \text{otherwise}. \end{cases} \tag{15}$$

The far reward encourages approaching the standing point:

$$r_t^{\text{far}} = 0.5\, r_t^{\text{walk}} + 0.2\, r_t^{\text{vel}} + 0.2\, r_t^{\text{facing}} + 0.1\, r_t^{\text{stand}}, \tag{16}$$

where r_t^{walk} rewards walking toward the standing point, r_t^{vel} aligns velocity, r_t^{facing} ensures proper facing direction, and r_t^{stand} rewards correct height.

The near reward focuses on lying accuracy:

$$r_t^{\text{near}} = 0.5\, r_t^{\text{pos}} + 0.3\, r_t^{\text{head}} + 0.2\, r_t^{\text{alignment}}, \tag{17}$$

where r_t^{pos} minimizes the distance to the target, r_t^{head} aligns head height, and $r_t^{\text{alignment}}$ rewards proper body alignment.

3.5. Prone Skill

Definition. The prone task requires the character to position its root joint at a designated 3D prone position on an object, typically centered on the object's surface. Unlike the lying task, the character must face downward while maintaining alignment with the target surface.

Task Observation. The observation $g_t^p \in \mathbb{R}^{35}$ includes the 3D target prone position $\in \mathbb{R}^3$, the 3D root position $\in \mathbb{R}^3$, the root rotation $\in \mathbb{R}^6$, the 2D front-facing direction $\in \mathbb{R}^2$, and the positions of eight corner points of the object's bounding box $\in \mathbb{R}^{3\times 8}$. These observations help guide the approach and ensure the correct orientation for prone positioning.

Task Reward. The prone reward r_t^p encourages the character to transition smoothly from moving to a prone position while maintaining proper alignment and facing downward. The reward is defined as:

$$r_t^p = \begin{cases} 0.7\, r_t^{\text{near}} + 0.3\, r_t^{\text{far}}, & \|x_t^{\text{root}} - x_t^{\text{tar}}\| > 1.5, \\ r_t^{\text{near}}, & \text{otherwise}. \end{cases} \tag{18}$$

The far reward encourages approaching the target prone position:

$$r_t^{\text{far}} = 0.5\, r_t^{\text{walk}} + 0.2\, r_t^{\text{vel}} + 0.2\, r_t^{\text{facing}} + 0.1\, r_t^{\text{height}}, \tag{19}$$

where r_t^{walk} rewards moving toward the prone position, r_t^{vel} aligns velocity with the direction of motion, r_t^{facing} ensures proper facing direction, and r_t^{height} encourages maintaining an appropriate height during approach.

The near reward focuses on prone accuracy:

$$r_t^{\text{near}} = 0.6\, r_t^{\text{pos}} + 0.2\, r_t^{\text{alignment}} + 0.2\, r_t^{\text{face\_down}}, \tag{20}$$

where r_t^{pos} minimizes the distance to the prone target, $r_t^{\text{alignment}}$ ensures proper body alignment with the surface, and $r_t^{\text{face\_down}}$ rewards the character for maintaining a face-down orientation.

3.6. Support Skill

Definition. The support task encourages the character to approach a target object and maintain stable interaction by placing its hands on the top surface while keeping stable foot placement and proper posture.

Task Observation. The task observation $g_t^m \in \mathbb{R}^{27}$ consists of the 3D target position of the object's top surface center ($x_t^o, z_t^o \in \mathbb{R}^3$) and the 3D coordinates of the eight corner points of the object's bounding box ($b_t \in \mathbb{R}^{3\times 8}$).

Task Reward. The total reward r_t^m is defined as:

$$r_t^m = \begin{cases} 0.4r_t^f + 0.6r_t^s, & \|x_t^o - x_t^r\| > 1.5, \\ r_t^s, & \text{otherwise}, \end{cases} \tag{21}$$

$$r_t^f = 0.5\exp\big(-0.5\|x_t^o - x_t^r\|^2\big) \tag{22}$$

$$+ 0.5\exp\big(-2.0\|1.5 - d_t^* \cdot \dot{x}_t^r\|^2\big), \tag{23}$$

$$r_t^s = 0.3r_t^h + 0.2r_t^g + 0.15r_t^t + 0.2r_t^o + 0.15r_t^z, \tag{24}$$

where r_t^f encourages the character to approach the object, and r_t^s combines five components for stable interaction:

$$r_t^h = 0.6\exp\big(-20\|z_t^h - z_t^o\|^2\big) \tag{25}$$

$$+ 0.4\exp\big(-5\|x_t^{h2} - x_t^o\|^2\big), \tag{26}$$

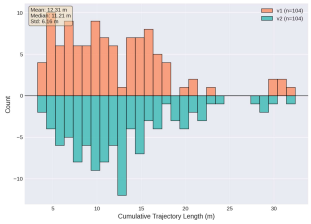
$$r_t^g = \exp\big(-50\|z_t^f - z_g\|^2\big), \tag{27}$$

$$r_t^t = \exp\big(-10\|x_t^{fr} - x_t^{fl}\|^2\big), \tag{28}$$

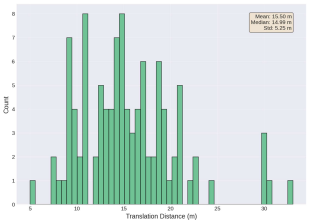
$$r_t^o = \exp\big(-2\|1.0 - (-u_t^b)\|^2\big), \tag{29}$$

$$r_t^z = \exp\big(-10\|z_t^r - z_t^o\|^2\big). \tag{30}$$

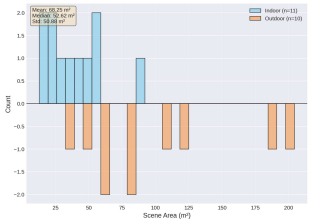
Here, x_t^o and x_t^r denote the 2D positions of the object and the character's root, while z_t^o and z_t^r are their respective heights. x_t^{h2} and z_t^h represent the 2D position and height of the hands. Similarly, x_t^{fr}, x_t^{fl}, and z_t^f refer to the 2D positions and height of the feet, z_g is the ground height, and $-u_t^b$ is the vertical component of the body's up direction.



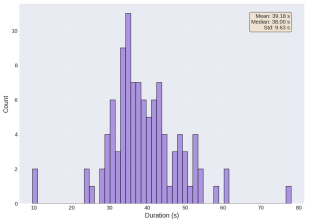
(a) Camera Trajectory Length Distribution.



(b) Human Trajectory Length Distribution.



(c) Scene Mesh Area Distribution.



(d) Sequence Length Distribution.

Figure 3. Statistical information of collected dataset.

Evaluation The evaluation of the Support task focuses on the agent's ability to position its hands on the top surface of the target object and keep its feet close together. The key metric is the combined XY-plane distance and Z-axis deviation between the hands and the object's top surface. The task is deemed successful if the hands are within predefined thresholds and the feet maintain adequate proximity for stability.

4. More Details of Scene-Aware Imitation Policy

4.1. Representations

Character Proprioception. The state s describes the proprioception of the character's body, with features consisting of the relative positions of each link with respect to the root (designated to be the pelvis), their rotations expressed in quaternions, and their linear and angular velocities. All features are computed in the character's local coordinate frame, with the root at the origin and the x-axis along the root link's facing direction.

Height Map. To perceive the surrounding scene geometry, we utilize a local egocentric height map. This map is structured as an 11×11 grid spanning a 2m $\times$ 2m area centered on the humanoid, resulting in a sampling interval of 0.2m. The grid is defined within the character's local coordinate frame; consequently, the sampling points dynamically translate and rotate with the humanoid's movement and heading, consistently covering the immediate vicinity. The height values at these grid points are queried from a high-resolution underlying scene mesh (0.05m resolution) using nearest-neighbor interpolation.

Target States. The target state $\hat{q}$ encodes the desired future motion of the character. It is constructed by sampling a short trajectory segment from the dataset spanning three consecutive future time steps: $T, T+1$, and $T+2$. For each time step, the state comprises the positions, rotations, linear velocities, and angular velocities of all body links. All features are transformed from the world frame into the simulated character's local coordinate frame. This local frame is defined with the character's root located at the origin and the x-axis aligned with the root link's facing direction.

Action. Our simulated humanoid is constructed based on the SMPL body model, comprising 23 controllable joints. Each joint possesses 3 degrees of freedom (DoF), and we employ a Proportional-Derivative (PD) controller for each DoF. Consequently, the action $a \in \mathbb{R}^{69}$ generated by the policy specifies the target orientations for these PD controllers.

4.2. Reward

To encourage the character to closely reproduce the reference motion while maintaining motion naturalness, our reward function r_t is composed of two terms: a tracking reward r_t^{track} and a jitter penalty r_t^{smooth}. The tracking reward incentivizes the policy to minimize the kinematic error between the simulated character and the reference motion. The jitter penalty is introduced to suppress abnormal shaking generated when the character interacts with objects, which may be induced by instabilities in the physics simulation. The total reward is defined as:

$$r_t = r_t^{\text{track}} - r_t^{\text{smooth}}. \tag{31}$$

Figure 4. **Rendered SMPL and depth images of the captured dataset in camera space.**

The tracking reward r_t^{track} is computed as the weighted sum of exponential differences across all humanoid links:

$$\begin{aligned} r_t^{\text{track}} = & \, w_{\text{jp}} \exp\left(-100\|\hat{\boldsymbol{p}}_t - \boldsymbol{p}_t\|^2\right) \\ & + w_{\text{jr}} \exp\left(-10\|\hat{\boldsymbol{q}}_t \ominus \boldsymbol{q}_t\|^2\right) \\ & + w_{\text{jv}} \exp\left(-0.1\|\hat{\boldsymbol{v}}_t - \boldsymbol{v}_t\|^2\right) \\ & + w_{\text{j}\omega} \exp\left(-0.1\|\hat{\boldsymbol{\omega}}_t - \boldsymbol{\omega}_t\|^2\right), \end{aligned} \tag{32}$$

where the equation penalizes the differences in translation $\boldsymbol{p}$, rotation $\boldsymbol{q}$, linear velocity $\boldsymbol{v}$, and angular velocity $\boldsymbol{\omega}$ for all rigid body links of the humanoid between the simulation and the reference. The jitter penalty penalizes the magnitude of the difference between consecutive actions, defined as:

$$r_t^{\text{smooth}} = \|\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\|^2, \tag{33}$$

where $\boldsymbol{a}_t$ and $\boldsymbol{a}_{t-1}$ denote the action at the current and previous time steps, respectively. By minimizing the rate of change of the actions, the policy is incentivized to generate continuous and stable control trajectories, thereby reducing jittery behaviors.

5. More Details of Captured Dataset Used in Main Paper

We collected data from 23 scenes, each with a high-precision mesh, 104 sequences, and approximately 200,000 video frames. Each frame is accompanied by corresponding depth maps, segmentation masks, camera trajectories, and human parameters(bounding boxes, 2D keypoints, SMPL parameters).

In Fig. 3a, we present the distribution of camera trajectory lengths, which range from 4 meters to over 30 meters. In Fig. 3b, the human trajectory length distribution is shown, with performers moving between 5 meters and over 30 meters. Figure 3c illustrates the scene mesh area

distribution. Indoor scenes are relatively smaller, ranging from 20 to 90 square meters, while outdoor scenes can be as large as 200 square meters. Finally, in Fig. 3d, we show the sequence length distribution, where most sequences have durations ranging from 30 to 60 seconds.

5.1. Qualitative Demonstrations

We show camera space results in Sec. 3.6 and world space results in Sec. 5.1

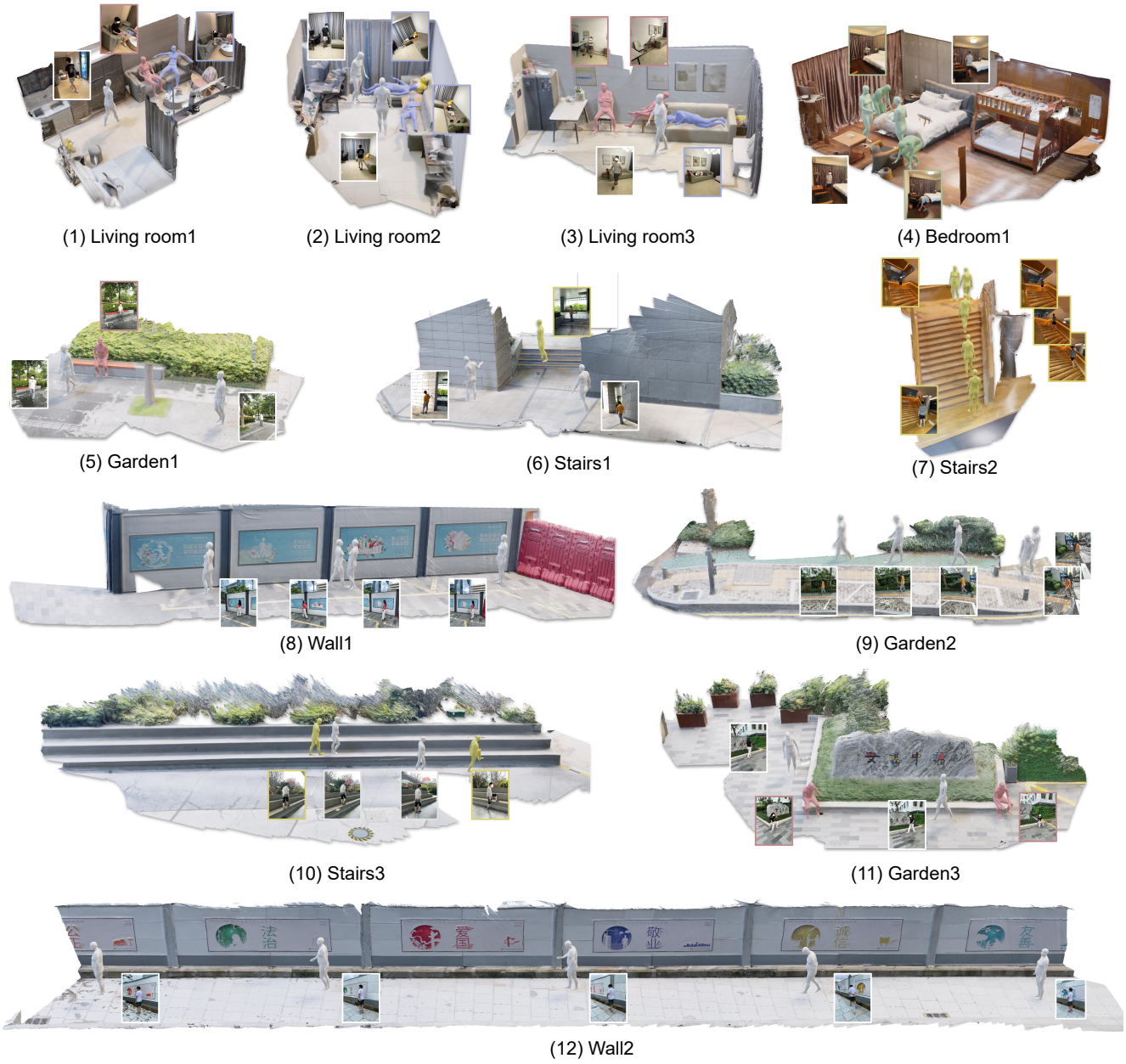


Figure 5. **3D demo of the captured dataset.**